Contention-based Learning MAC Protocol for Broadcast Vehicle-to-Vehicle Communication

[1]Andreas Pressas, [1]Zhengguo Sheng, [1]Falah Ali, [2]Daxin Tian, [1]Maziar Nekovee

[1]Department of Engineering and Design, University of Sussex, UK
[2]School of Transportation Science and Engineering, Beihang University, China
Email: a.pressas@sussex.ac.uk

***Abstract*—Vehicle-to-Vehicle Communication (V2V) is an upcoming technology that can enable safer, more efficient transportation via wireless connectivity among moving cars. The key enabling technology, specifying the physical and medium access control (MAC) layers of the V2V stack is IEEE 802.11p, which belongs in the IEEE 802.11 family of protocols originally designed for use in WLANs. V2V networks are formed on an ad hoc basis from vehicular stations that rely on the delivery of broadcast transmissions for their envisioned services and applications. Broadcast is inherently more sensitive to channel contention than unicast due to the MAC protocol's inability to adapt to increased network traffic and colliding packets never being detected or recovered.**

This paper addresses this inherent scalability problem of the IEEE 802.11p MAC protocol. The density of the network can range from being very sparse to hundreds of stations contenting for access to the channel. A suitable MAC needs to offer the capacity for V2V exchanges even in such dense topologies which will be common in urban networks. We present a modified version of the IEEE 802.11p MAC based on Reinforcement Learning (RL), aiming to reduce the packet collision probability and bandwidth wastage. Implementation details regarding both the learning algorithm tuning and the networking side are provided. We also present simulation results regarding achieved message packet delivery and possible delay overhead of this solution. Our solution shows up to 70% increase in throughput compared to the standard IEEE 802.11p as the network traffic increases, while maintaining the transmission latency within the acceptable levels.

I. Introduction

V2V technology aims to enable safer and more sophisticated transportation starting with minor, inexpensive additions of communication equipment on conventional vehicles and moving towards network-assisted fully autonomous driving. It will be a fundamental component of the Intelligent Transportation Services and the Internet of Things (IoT). This technology allows for the formation of Vehicular Ad Hoc Networks (VANETs), a new type of network which allows the exchange of kinematic data among vehicles, for the primary purpose of safer and more efficient driving, as well as efficient traffic management and other third-party services. VANETs can help minimise road accidents and randomness in driving with on-time alerts, as well as enhance the whole travelling experience with new infotainment systems which allow acquiring navigation maps and other information from peers.

The V2V radio technology is based on the IEEE 802.11a stack, adjusted for low overhead operations in the Dedicated Short Range Communications (DSRC) spectrum (30 MHz in the 5.9 GHz band for Europe). It is being standardised as IEEE 802.11p [1]. The adjustments that have been made are mainly for enabling exchanges without belonging in a Basic Service Set. Consequently, communication via IEEE 802.11p is not managed by a central access point (as in typical Wireless LANs) but is ad hoc in nature.

A. Motivation

VANETs are the first large scale network to operate primarily on broadcast transmissions, since the data exchanges are often relevant for vehicles within an immediate geographical Region of Interest (ROI) of the host vehicle. This allows the transmission of broadcast packets (packets not addressed to a specific MAC address), so that they can be received from every vehicle within range without the overhead of authentication and association with an access point. Broadcasting has always been controversial for the IEEE 802.11 family of protocols [2], since they treat unicast and broadcast frames differently. Radio signals are likely to overlap with others in a geographical area and two or more stations will attempt to transmit using the same channel, leading to contention. Broadcast transmissions are inherently unreliable and more prone to contention, since the MAC specification in IEEE 802.11 does not request explicit acknowledgements (ACK packet) on receipt of broadcast packets, to avoid the ACK storm phenomenon, which appears when all successful receivers attempt to send back an ACK simultaneously and consequently congest the channel. This has not changed in the IEEE 802.11p amendment.

A MAC protocol is part of the data link layer (L2) of the OSI model and defines the rules of how the various network stations share access to the channel. The de-facto MAC layer used in IEEE 802.11-based networks is called Carrier Sense Multiple Access (CSMA) with Collision Avoidance (CA) protocol. It is a simple decentralized contention-based access scheme which has been extensively tested in WLANs and mobile ad hoc networks. The IEEE 802.11p stack also employs the classic CSMA/CA MAC. Although the proposed stack works fine for sparse VANETs with few nodes, it quickly shows its inability to accommodate increased network traffic because of the lack of ACKs. The lack of ACKs not only makes transmissions unreliable, but also does not provide any feedback mechanism for the CSMA/CA backoff mechanism. So it cannot adapt and resolve contention among stations when the network is congested.

The DSRC operation requires that L1 and L2 must be built in a way that they can handle a large number of contenting nodes in the communication zone, on the order of 50 to 100. The system should not collapse from saturation if this number is exceeded. Intel's Automated Driving Solutions Division [3] predict that autonomous cars will each generate approximately 4 terabytes of data in about an hour and a half of driving or the amount of time a typical person spends in their car each day. The data is divided into technical (i.e., vehicular, proximity sensors, radars), crowd-sourced (i.e., maps, environment, traffic, parking) and personal (i.e., VoIP, Internet radio, routes) applications. We believe that a significant part of this data will be exchanged through V2V links, making system scalability a critical issue to address. There is a need for an efficient MAC protocol for V2V communication purposes, that adapts to the VANET's density and transmitted data rate, since such network conditions are not known a-priori.

B. Contribution

In this paper, we propose to apply Reinforcement Learning (RL) [4] in the context of medium access control for broadcast wireless communication in vehicular environments. Machine Learning-based techniques have the potential to enter and improve every layer of the network stack for the IoT and other applications. RL is a general class of machine learning algorithms fit for problems of sequential decision making and control. It can be used as a parameter-perturbation/adaptive-control method for Markov Decision Processes (MDPs) [5], a discrete time stochastic control formulation. RL is based on the idea that if an action is followed by a satisfactory state of affairs, or by an improvement in the state of affairs (or a reward function), then the agent's tendency to produce that action is strengthened, i.e., reinforced. Specifically, we develop and evaluate a solution based on Q-Learning, a much-used model-free RL algorithm that can solve MDPs with very little information from the dynamic VANET environment but still reveals effective solutions regarding contention control for various network conditions. In addition, we employ a strategy for building self-improving Q-Learning controllers that yield instant performance benefits since the vehicle-station's deployment and always strive for optimum operation while on-line.

The remainder of the paper is outlined as follows: Section II reviews related work. Section III briefly reviews the IEEE 802.11p MAC protocol for broadcast communication. Section IV presents the Reinforcement Learning model. The RL-based MAC protocol for V2V broadcast communication is developed in Section V. In Section VI, we evaluate the proposed protocol's performance in comparison to the IEEE 802.11p standard. Section VII presents our conclusions.

II. Related Work

Related work in [6] shows that IEEE 802.11p exhibits lower latency and higher delivery ratio than LTE in scenarios fewer than 50 vehicles. More specifically, for smaller network densities, the standard allows end-to-end delays less than 100 ms and throughput of 10 kbps which satisfies the requirements set by active road safety applications and few of the lightweight cooperative traffic awareness applications. However, as the number of vehicles increases, the standard is unable to accommodate the increased network traffic and support performance requirements for more demanding applications.

When it comes to work which is focused specifically on the MAC layer issues, [7] uses the Markov Decision Process (MDP) formulation to design a MAC with deterministic backoff for virtualized IEEE 802.11 WLANs. For V2V exchanges, the work presented in [8] examines the IEEE 802.11p MAC regarding channel contention using the Markov model from [9] and proposes a passive contention estimation technique by observing the count of idle inter-frame slots.

RL is inspired by behaviourist psychology and deals with how software agents should take actions in an environment while aiming to maximize their cumulative reward. The problem because of its generality, is studied in many disciplines, such as game theory, control systems, IT, simulation-based optimization, statistics, and genetic algorithms. There have been attempts to apply RL for optimizing the access control layer of wireless networks. The protocol in [10] is targeted on wireless sensor networks, optimising battery-power node energy consumption. The protocol in [11] is targeted on wireless vehicular networks that operate on a unicast basis. It employs contention window adaptation [12] which is a proven technique to improve the network contention because of interference in wireless networks. The premise is interesting, but the proposed IEEE 802.11p is a broadcast-based protocol. The current literature does not deal with the broadcasting issues within the context of contention resolution on the MAC level. VANETs will be the first large-scale networks to operate primarily on broadcast exchanges.

III. The IEEE 802.11p Medium Access Control

The MAC protocol is responsible for transferring data securely when there is more than one station attempting to access the same channel simultaneously. An efficient MAC will strive for maximum channel utilization with minimum collisions. The Distributed Coordination Function (DCF) is the fundamental MAC technique of the IEEE 802.11-based WLAN standards. DCF employs the CSMA/CA algorithm.

A. CSMA/CA algorithm

We start with the basic principle of the medium access operation for IEEE 802.11-based networks, which works as follows:

- Once a packet is ready for transmission, the station is required to sense the state of the wireless medium before transmitting (listen before talk) to determine whether another station is transmitting or not.
- If it finds that the medium is continuously idle for a DCF Interframe Space (DIFS) period, the station is given permission to transmit after it goes through an additional time period called backoff. The purpose of the backoff is to introduce some asynchronisation which helps in

the case two station's DIFS expire simultaneously. When the backoff counter reaches 0, the packet is transmitted immediately.

- If the channel turns busy during the DIFS interval, the node defers from transmission until the medium is again found idle for the duration of a DIFS interval.
- When a unicast packet has been received correctly, the destination node waits for a Short Interframe Space (SIFS) interval, to give priority to an ACK packet sent back to the source node immediately after the reception has finished to confirm successful reception.

B. Backoff Mechanism

The range of the generated random backoff timer is bounded by the contention window. More especially, the node randomly draws an integer b from the uniform distribution over the interval $[0, CW]$, where the initial CW value equals CW_{min}, and counts down for b time slot intervals before attempting to transmit. The backoff value will be reduced only when the channel is free, or else the counter freezes until the medium turns idle again.

In the classic IEEE 802.11-based unicast networks, the CW parameter adapts to a value between CW_{min} and CW_{max}, depending on the delivery outcome of the transmitted packets. If a packet transmission fails (ACK not received), the CW parameter is doubled. If the following transmission fails, the CW is doubled again and so goes on until it successfully transmits a packet and receives an ACK, so it resets CW to CW_{min}, or fails until it reaches CW_{max}. By using this mechanism it is less probable that two or more nodes pick the same b value and transmit simultaneously.

C. MAC-level Broadcasting

In broadcast transmissions, though, which is the primary way of exchanging information in IEEE 802.11p-based networks, there is no reaction to increases in network load by enlarging the CW. The reason for this is that original packets are not acknowledged to avoid the acknowledgement storm problem, because every recipient would invoke a SIFS interval and try to send back an ACK, which would cause interference and lead to collisions. Consequently, for the broadcasting case, the backoff counter b reinitialises to a uniformly distributed value within $[0, CW_{min}]$ no matter the outcome of the attempted transmission. The operation of CSMA/CA for both unicast and broadcast transmissions can be seen in Fig. 1.

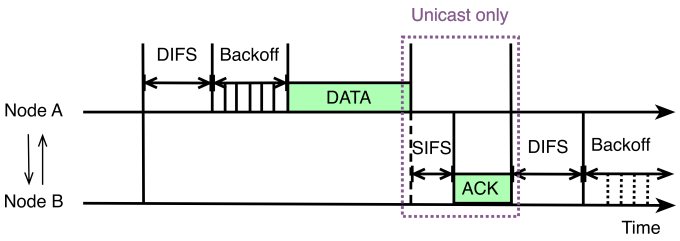

Fig. 1. A CSMA/CA cycle for both unicast and broadcast cases. It manages channel access among transmitting nodes A and B.

IV. Q-Learning in Markovian Environments

A. Markov Decision Processes

In RL, the learning agents can be studied mathematically by adopting the MDP formalism. An MDP is defined as a (S, A, P, R) tuple, where S stands for the set of possible states, A_s is the set of possible actions from state $s \in S$, $P_a(s, s')$ is the probability to transit from a state $s \in S$ to $s' \in S$ by performing an action $a \in A$. $R_a(s, s')$ is the reinforcement (or immediate reward), result of the transition from state s to state s' because of an action a, as seen in Fig. 2. The decision policy π maps the state set to the action set, $\pi : S \rightarrow A$. Therefore, the MDP can be solved by discovering the optimal policy that decides the action $\pi(s) \in A$ that the agent will make when in state $s \in S$.

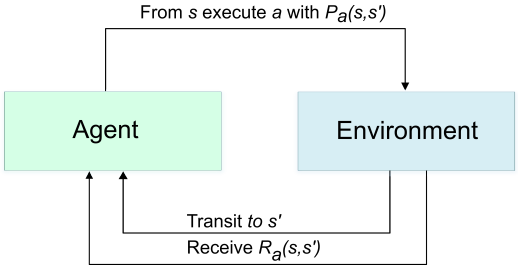

Fig. 2. Absract MDP model

B. Q-Learning

There are, though, many practical scenarios, such as the channel access control problem studied in this work, for which the transition probability $P_{\pi(s)}(s, s')$ or the reward function $R_{\pi(s)}(s, s')$ are unknown, which makes it difficult to evaluate the policy π. Q-learning [13] [14] is an effective and popular algorithm for learning from delayed reinforcement to determine an optimal policy π in absence of the transition probability. It is a form of model-free reinforcement learning which provides agents the ability to learn how to act optimally in Markovian domains by experiencing the consequences of their actions, without requiring maps of these domains.

In Q-learning, the agent maintains a table of $Q[S, A]$, where S is the set of states and A is the set of actions. At each discrete time step $t = 1, 2, \ldots, \infty$, the agent observes the state $s_t \in S$ of the MDP, selects an action $a_t \in A$, receives the resultant reward r_t and observes the resulting next state $s_{t+1} \in S$. This experience (s_t, a_t, r_t, s_{t+1}) updates the Q-function at the observed state-action pair, thus provides the updated $Q(s_t, a_t)$. The algorithm, therefore, is defined by the function (1) that calculates the quantity of a state-action (s, a) combination. The goal of the agent is to maximise its cumulative reward. The core of the algorithm is a value iteration update. It assumes the current value and makes a correction based on the newly acquired information, as in (1).

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \times [r_t + \gamma \times \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) \; - Q(s_t, a_t)], \tag{1}$$

where the discount factor γ models the importance of future rewards. A factor of $\gamma = 0$ will make the agent "myopic" or short-sighted by only considering current rewards, while a factor close to $\gamma = 1$ will make it strive for a high long-term reward. The learning rate α quantifies to what extent the newly acquired information will override the old information. An agent with $\alpha = 0$ will not learn anything, while with $\alpha = 1$ it would consider only the most recent information. The $\max_{a_{t+1} \in A} Q(s_{t+1}, a_{t+1})$ quantity is the maximum Q value among possible actions in the next state. In the following sections we present employing (1) as a learning, self-improving, control method for managing channel access among IEEE 802.11p stations.

V. Proposed MAC Protocol

The adaptive backoff problem fits into the MDP formulation. RL is used to design a MAC protocol that selects the appropriate CW parameter based on gained experience from its interactions with the environment within an immediate communication zone. The proposed MAC protocol features a Q-Learning-based algorithm that adjusts the contention window size based on binary feedback from probabilistic rebroadcasts in order to avoid packet collisions.

A. The Action Selection Dilemma

The state space S contains the discrete IEEE 802.11p-compatible CW values ranging from $CW_{min} = 3$ to $CW_{max} = 255$. The CW is adapted prior to every packet transmission by performing one of the following actions.

$$CW_{t+1} \xleftarrow{a \in \{CW_t - 1/2, CW_t, CW_t * 2 - 1\}} CW_t \,. \tag{2}$$

RL differs from supervised learning in that correct input/output pairs are never presented, nor sub-optimal actions are explicitly corrected. In addition, in RL there is a focus on on-line performance, which involves finding a balance between exploration of uncharted territory and exploitation of current knowledge. This in practice translates as a trade-off in how the learning agent in this protocol selects its next action for every algorithm iteration. It can either randomly pick an action from (2) (exploration) so that the algorithm can transit to a different (s, a) pair and get experience (reward) for it, or follow a greedy strategy (exploitation), and choose the action with the highest Q-value for its current state given by

$$\pi(s) = \arg\max_a Q(s, a) \,. \tag{3}$$

B. Convergence Requirements

The RL algorithm's purpose is to converge to a (near) optimum output, in terms of CW. Watkins and Dayan [13] proved that Q-Learning converges to the optimum action-values with probability 1 as long as all actions are repeatedly sampled in all states and the action-value pairs are represented discretely.

The greedy policy with respect to the Q-values tries to exploit continuously, however, since it does not explore all (s, a) pairs properly, it fails satisfying the first criterion. At the other extreme, a fully random policy continuously explores all (s, a) pairs, but it will behave sub-optimally as a controller. An interesting compromise between the two extremes is the ε-greedy policy [4], which executes the greedy policy with probability $1 - \varepsilon$. This balancing between exploitation and exploration can guarantee convergence and often good performance.

The proposed protocol uses the ε-greedy strategy to focus the algorithm's exploration on the most promising CW trajectories. Specifically, it guarantees the first convergence criterion by forcing the agent to sample all (s, a) pairs over time with probability ε. Consequently, the proposed algorithmic implementation satisfies both convergence criteria, but further optimisation is needed regarding convergence speed and applicability of the system.

In practice the Q-Learning algorithm converges under different factors depending on the application and complexity. When deployed in a new environment, the agent should mostly explore and value immediate rewards, and then progressively show its preference for the discovered (near) optimal actions $\pi(s)$ as it is becoming more sure of its Q estimates. This can be achieved via the decay function shown in (4).

$$\varepsilon = \alpha = 1 - \frac{N_{tx}}{N_{decay}} \quad \text{for} \quad 0 \leq N_{tx} \leq N_{decay}, \tag{4}$$

where N_{tx} is the number of transmitted broadcast packets and N_{decay} is a pre-set number of packets that sets the decay period. This decay function is necessary to guarantee convergence towards the last known optimum policy in probabilistic systems such as the proposed contention-based MAC, since there is no known optimum final state. By reducing the values of ε and α over time via (4), the agent is forced to progressively focus on exploitation of gained experience and strive for a high long term reward. This way, when approaching the end of the decay period the found (near) optimal states-CW/s are revealed.

C. A-Priori Approximate Controller

The above strategy can be used to get instant performance benefits, starting from the first transmission. This is done by pre-loading approximate controllers, pre-trained for different transmitted bit rates and number of neighbours via (4), to the station's memory. These controllers define an initial policy that positively biases the search and accelerates the learning process.

The agent's objective in this phase is to quickly populate its Q-table with values (explore all the state-action pairs multiple times) and form an initial impression of the environment. The lookup Q-table is produced by encoding this knowledge (Q-values) for a set period of N_{decay} a priori and can be used as an initial approximate controller which yields an instant performance benefit since the system is deployed.

Q-learning is an iterative algorithm so it implicitly assumes an initial condition before the first update occurs. Zero initial conditions are used the very first time the algorithm is trained

on a set environment, except from some forbidden state-action pairs with large negative values, so it does not waste iterations in which it would try to increase/decrease the CW when it is already set on the upper/lower limit. The algorithm is also explicitly programmed to avoid performing these actions on exploration. The un-trained, initial Q-table is set as in (5), where the rows represent the possible states - CW sizes and columns stand for the action space.

$$\mathbf{Q_0}[\mathbf{7}][\mathbf{3}] = \begin{array}{c} \\ \mathbf{3} \\ \mathbf{7} \\ \mathbf{15} \\ \mathbf{31} \\ \mathbf{63} \\ \mathbf{127} \\ \mathbf{255} \end{array} \begin{array}{c} \begin{array}{ccc} \mathbf{CW} & {}^{(CW\,-\,1)}/_{2} & CW \qquad CW*2+1 \end{array} \\ \begin{pmatrix} -100 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -100 \end{pmatrix} \end{array} \tag{5}$$

We propose that each station employs different learning, self-improving, controllers and uses the appropriate one depending on a combination of sensed density and received bit rate. This is feasible because the station has the ability to sense the number of one-hop neighbours since they all transmit heart-beat, status packets periodically. It also does not have the memory constraints that typical sensor networks have. An example of a controller's table at the end of the ε decay period as in (4) can be seen in (6). The controller has been trained a-priori with $\gamma = 0.7$ and a decay period lasting for 180 s in a 60-car network, where every car transmits 256 bytes every 100 ms. A trajectory leading to optimum/near-optimum CW/s is being formed (depending on past experience) by choosing the maximum Q-value for every CW-state, seen in bold font. The controller in (6) oscillates between the values 31 and 63 when exploiting the Q-table to find the optimum CW.

$$\mathbf{Q}^{\pi}[\mathbf{7}][\mathbf{3}] \approx \begin{array}{c} \\ \mathbf{3} \\ \mathbf{7} \\ \mathbf{15} \\ \mathbf{31} \\ \mathbf{63} \\ \mathbf{127} \\ \mathbf{255} \end{array} \begin{array}{c} \begin{array}{ccc} \mathbf{CW} & {}^{(CW\,-\,1)}/_{2} & CW \qquad CW*2+1 \end{array} \\ \begin{pmatrix} -100 & -0.07218 & \mathbf{0.2388} \\ -0.076 & -0.0325 & \mathbf{0.6748} \\ 0.198 & 0.28012 & \mathbf{0.817} \\ 0.2896 & 0.2985 & \mathbf{0.4917} \\ \mathbf{0.4945} & 0.10115 & 0.2838 \\ \mathbf{0.2043} & -0.055 & -0.0218 \\ \mathbf{0.1745} & -0.86756 & -100 \end{pmatrix} \end{array} \tag{6}$$

D. On-line Controller Augmentation

While the pre-trained, pre-loaded, approximate controller is useful for speeding up the learning process as well as getting an instant performance benefit, its drawback is that by default it is not adaptive to change in the environment while on-line. The on-line efficiency of the Q-Learning controller depends on finding the right balance between exploitation of the station's current knowledge, and exploration for gathering new information. This means that the algorithm must sometimes perform actions other than the ones dictated by the current policy, to update and augment that controller with new information.

While the station is on-line, exploratory action selection is performed less frequently ($\varepsilon = 0.1$) than in a-priori learning (4) (ε starts from 1), primarily to compensate for modelling errors in the approximate controller. This means that the controller in its on-line operation uses the optimum Q-value 90% of the time, and makes exploratory CW perturbations 10% of the time in order to gain new experience. In this way the agent still has the opportunity to correct its behaviour based on new interactions with the VANET and corresponding rewards.

E. Implementation Details

In RL, the only positive or negative reinforcement an agent receives upon acting so that it can learn to behave correctly in its environment, comes in a form of a scalar reward signal. Taking advantage of the link capacity for maximum packet delivery (throughput) was of primary concern for this design, aiming to satisfy the requirements of V2V traffic (frequent broadcasting of kinematic and multimedia information). For this purpose, the reward function is based on the success of these transmissions. Reward r can be either 1 or -1 for successful (ACK) and failed transmissions (no ACK) correspondingly. A successful transmission from the same consecutive state - CW is not given any reward. The following pseudo-code summarizes our proposed protocol.

Algorithm 1 Q-Learning V2V MAC

1: Initialize $Q_0(CW, A)$ at $t_0 = 0$ ▷ as in (5)
2: **procedure** ACTION-SELECTION(CW_t) ▷ ε-greedy
3: **if** $p_\varepsilon \leq \varepsilon$ **then**
4: $a_{t+1} \leftarrow random[{}^{(CW_t\,-\,1)}/_{2}, CW_t, CW_t * 2 - 1]$
5: **else if** $p_\varepsilon \geq 1 - \varepsilon$ **then**
6: $a_{t+1} \leftarrow a_\pi$ ▷ Optimum a from (3)
7: **end if**
8: **if** A-priori Controller Learning **then**
9: $\varepsilon = \alpha \rightarrow decay$ ▷ according to rule (4)
10: **else if** On-line Learning **then**
11: $\varepsilon = \alpha \rightarrow constant$
12: **end if**
13: $CW_{t+1} \leftarrow CW^{a_{t+1}}$
14: **end procedure**
15: TX Broadcast Packet: $MessageId$ ▷ Transmit
16: **procedure** FEEDBACK(CW_{t+1}, a_{t+1}) ▷ Collect Reward
17: Initialize: $RTT \leftarrow 0$ s
18: **if** RX $MessageId$ AND $RTT < 0.1$ s **then**
19: **if** $a_t \neq (CW_{t+1} \longleftarrow CW_t)$ **then**
20: $r_t \leftarrow 1$
21: **end if**
22: **else if** $RTT \geq 0.1$ s **then**
23: $r_t \leftarrow -1$
24: **end if**
25: **end procedure**
26: Update $Q(CW_{t+1}, a_{t+1})$ ▷ according to rule (1)
27: GOTO 2

The first step of the MAC protocol would be to set the default CW of the station to the minimum possible value, which is suggested by the IEEE 802.11p standard. After

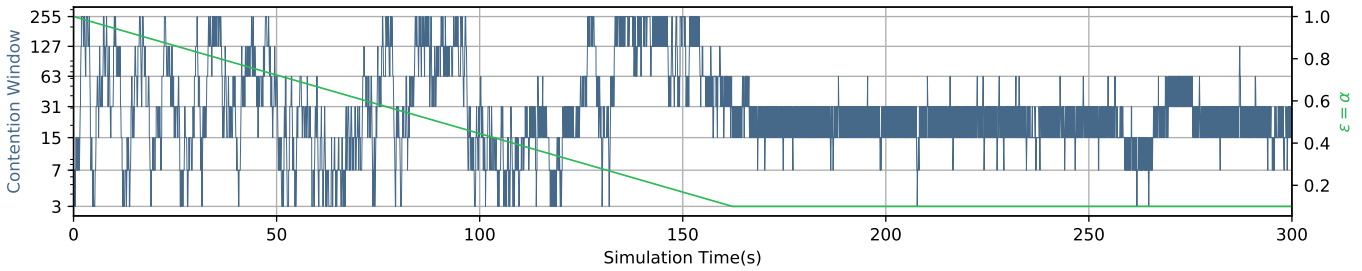

Fig. 3. Trace of CW over time for a station. The first stage is the a-priori controller training phase via (4) for 3 minutes (or $N_{decay} = 1800$ packets), then on-line stage for 2 minutes, with an exploration to exploitation ratio of 1:9

that the node makes an exploratory move with probability ε (exploration) or picks the best known action to date (highest Q value) with probability $1 - \varepsilon$.

We use packet rebroadcasts as ACKs, since some will be overheard from the source vehicle, even assuming that they move at the maximum speed limit. These rebroadcasts can happen for forwarding purposes and they enhance the reliability of the protocol, since the original packet senders can detect collisions, as well as provide a means to reward if they succeed in successfully broadcasting a packet. We use probabilistic rebroadcasting for simplicity, but various routing protocols can be used instead.

Every time a packet containing original information is transmitted, a timer is initiated which waits for a predefined time for an overheard retransmission of that packet, which will have the same $MessageId$. These broadcast packets are useful for a short lifetime, which is the period between refreshes. So a rebroadcast packet received after that period, is not considered to be a valid ACK because the information will not be relevant any more, since the nodes in VANETs attempt to broadcast fresh information frequently (i.e., 1-10 Hz).

VI. Performance Evaluation

The medium access control (MAC) method of the vehicular communication standard IEEE 802.11p has been simulated in a realistic vehicular traffic scenario with vehicle-stations periodically broadcasting packets. In order to evaluate the performance of our novel proposed protocol in comparison to the IEEE 802.11p protocol, simulations were carried out using the latest version of the OMNeT++ simulator and the Veins framework. Realistic mobility simulation is achieved by using SUMO coupled with the OMNeT++ stack.

A. Simulation Setup

All the cars within the area content for access to medium when trying to transmit a packet or rebroadcast a copy of one. Retransmission probability is set so that a proportion of nodes in the area of interest will rebroadcast the same information upon receipt (i.e., for 100 cars it is set at 2%). We collect most of our results within a specific ROI of $\sim$ 600 m x 500 m within the University of Sussex campus, and we set the power to a high enough level within the DSRC limit, in order to not be influenced by border effects (hidden/exposed terminals). The artificial campus map used for simulations can be seen in Fig. 4.

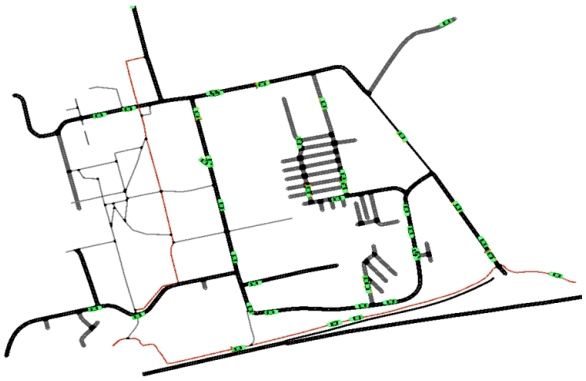

Fig. 4. Campus map used in network simulations

The achieved improvement on link-level contention was of primary concern, so a multitude of tests were run for a single hop scenario, with every node being within the range of the others. By eliminating the hidden terminal problem from the experiment and setting an infinite queue size, packet losses from collisions can be accurately measured. A multi-hop scenario is also presented, which makes the hidden terminal effect apparent in the performance of the network.

The simulation run time for the proposed MAC protocol consists of two stages, as seen in Fig.3. First is the approximate controller training stage, which lasts for $N_{decay} = 1800$ transmitted packets (or 180 s with $f_b = 10$ Hz). Then follows the evaluation or on-line period which lasts for 120 s, in which the agent acts with an $\varepsilon = \alpha = 0.1$. During this time, we benchmark the effect of the trained controllers regarding network performance as well as keep performing some learning for the controller augmentation. For IEEE 802.11p simulations, only the evaluation stage is needed, which lasts for the same time.

All cars in the network are continuously transmitting broadcast packets, such as Cooperative Awareness Messages (CAMs) or Decentralized Environmental Notification Messages (DENMs), with a period $T_b = \frac{1}{f_b} = 100$ ms. The packets are transmitted using the highest priority, voice traffic (AC_VO) access category. In VANETs, the network density changes depending on location and time of the day. We test

Parameter	Value
Evaluation time	120 s
A-priori training time	180 s
Channel frequency	5.9 Ghz
Transmission rate	6 Mbps
Transmission power	1-hop: 100 mW, 2-hop: 40 mW
Packet size L_p	256 bytes
Backoff slot time	13 μs
Broadcasting Frequency f_b	10 Hz
No of relays	$\geq$ 2 cars (probabilistic)
Discount rate γ	0.7
Learning rate α	training: eq. (4), on-line: 0.1
Epsilon ε	training: eq. (4), on-line: 0.1

TABLE I
SIMULATION PARAMETERS

the performance of the novel MAC against the standard IEEE 802.11p protocol for different number of cars. The data rate is set at 6 Mbps so it can conveniently accommodate hundreds of vehicles within the DSRC communication range.

B. Effect of Increased Network Density

We evaluate the scalability of the MAC protocols for a varying number of vehicles travelling in the simulated map described previously. The packet size L_p used in this scenario is 256 bytes, and the broadcasting frequency f_b is set at 10 Hz. Fig. 5 shows the increase of successfully delivered packets when using our novel MAC protocol. When using the standard IEEE 802.11p, packet delivery ratio (PDR) decreases in denser networks due to the increased collisions between data packets. The proposed MAC is designed to adjust the size of CW as needed to achieve maximum packet delivery.

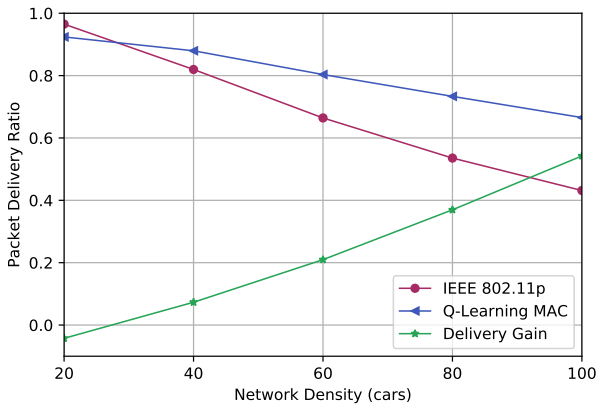

Fig. 5. PDR versus network density for broadcasting of 256-byte packets with $f_b = 10$ Hz

The PDR for the proposed Q-Learning MAC is measured after the initial, more exploratory phase (after the agent has gained some experience). We observed a 37.5% increase in performance (packets delivered) in a network formed of 80 cars when using the modified, "learning" MAC. There is a slight loss in performance (4%) for 20-car networks. In such sparse networks, the minimum CW is optimal, since with a big CW (waiting for more b time slots), transmission opportunities can be lost and the channel access delay will increase. When using our learning protocol, the agent still explores larger CW levels 10% of the time ($\varepsilon = 0.1$), for better adaptability and augmentation of it's initial controller. When the network density exceeds 40 cars, the proposed learning MAC performs much better regarding successful deliveries.

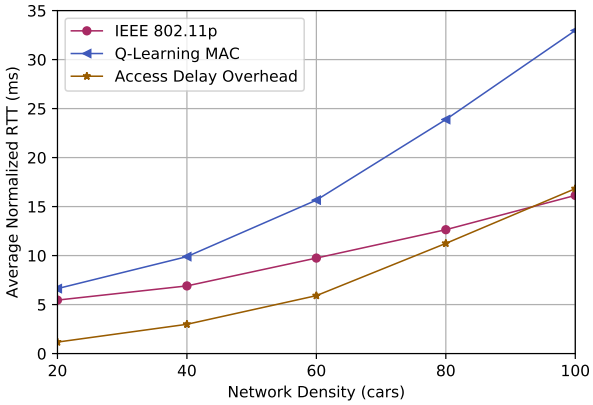

Fig. 6. Packet Return Time (delay) versus network density for broadcasting of 256-byte packets with $f_b = 10$ Hz

The Round-trip time (RTT) shown in Fig. 6 is defined as the length of time it takes for an original broadcast packet to be sent plus the length of time it takes for a rebroadcast of that packet to be received by the original sender. We can see that the increased CW of the learning MAC adds to the channel access delay time. The worse case scenario simulated is for 100 simultaneous transceivers within the immediate range of each other, in which the average RTT doubles to 32.8 ms when using the Q-Learning MAC. Given that both the transmission and heard retransmission are of the same packet size, we can assume that the mean delivery latency is 16.4 ms when using the learning MAC instead of 8 ms for baseline IEEE 802.11p, while PDR is improved by 54% .

C. Effect of Data Rate

We also examine the performance of both the standard and enhanced protocol for different data rates. PDR is measured for a network of 60 nodes without hidden terminals. The broadcasting frequency is set at $f_b = 10$ Hz, and the packet size L_p varies from 64 bytes to 512 bytes, as seen in Fig. 7. For 512 byte packets the mean achieved throughput T_{avg} per IEEE 802.11p node from (7) is 16.925 kbps. For the same settings, the learning MAC stations each achieve 29.218 kbps on average, yielding to a 72.63% increase in throughput. It is clear that for larger packet transmissions the Q-Learning based protocol will be much faster and more reliable.

$$T_{avg} = L_p \times f_b \times 8\,bit \times PDR\,. \tag{7}$$

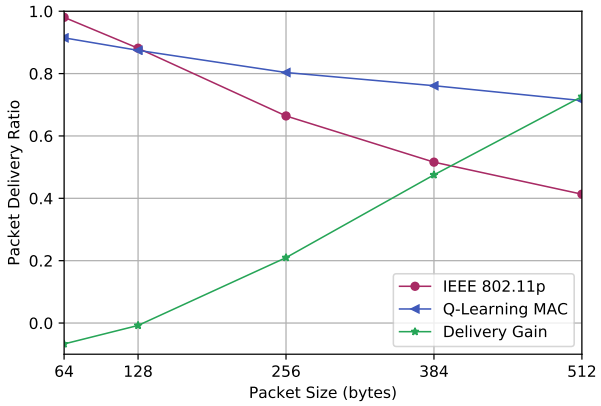

Fig. 7. PDR versus packet size for 60 vehicles broadcasting with $f_b = 10\,\text{Hz}$

D. Effect of Multi-hop

In a network without fixed topology, the most common way to disseminate information is to broadcast packets across the network. In VANETs, vehicles often cooperate to deliver data messages through multi-hop paths, without the need of centralized administration. In this scenario we test the performance of the proposed protocol when attempting to transmit two hops away. We evaluate performance for two-hop transmissions by reducing the transmission power to 40 mW. As the network density increases, the proposed MAC offers a valid delivery benefit for vehicle-stations contenting for access on the same channel. The performance of both IEEE 802.11p and the proposed learning MAC regarding two-hop packet reception ratio is shown in Fig. 8.

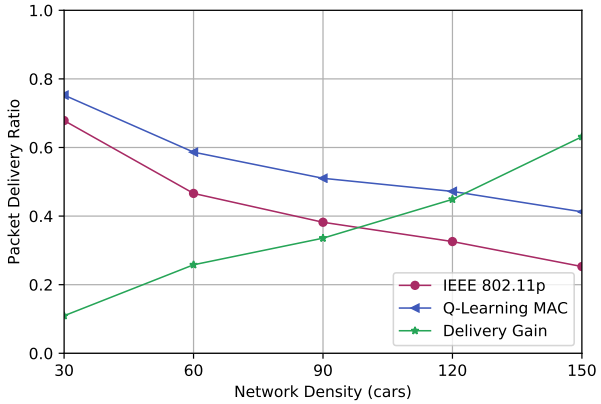

Fig. 8. PDR versus network density for broadcasting of 256-byte packets with $f_b = 10\,\text{Hz}$ in a two-hop scenario

We see that because the hidden terminal phenomenon appears the performance deteriorates compared to the single hop scenario, but the performance gain regarding packet delivery is still apparent when using Q-Learning to adapt the backoff. Packets lost are not recovered since we are concerned with the performance of the link layer.

VII. Conclusion

We have introduced a contention-based MAC protocol for V2V broadcast transmissions that relies on Q-Learning to discover the optimum contention window by continuously interacting with the network. We developed simulations to demonstrate the effectiveness of our MAC protocol. Results prove that the proposed protocol allows the network to scale better to increased network density and accommodate higher data rates compared to the IEEE 802.11p standard. This translates to more reliable packet delivery and higher system throughput, while maintaining acceptable delay levels. Future studies will be focused on how the learning MAC responds to drastic changes in the networking environment via invoking the ε decay function while on-line, as well as improving fairness and transmission latency.

Acknowledgement

This research was sponsored by The Engineering, and Physical Sciences Research Council (EPSRC) (EP/P025862/1) and Royal Society-Newton Mobility Grant (IE160920).

References

[1] "Ieee standard for information technology–telecommunications and information exchange between systems–local and metropolitan area networks–specific requirements part 11:wireless lan medium access control (mac) and physical layer (phy) specifications amendment 6: Wireless access in vehicular environments," ieee, pp. 1–51.." https://www.ietf.org/mail-archive/web/its/current/pdfqf992dHy9x.pdf, 2010.

[2] R. Oliveira, L. Bernardo, and P. Pinto, "The influence of broadcast traffic on ieee 802.11 dcf networks," *Computer Comm.*, vol. 32, no. 2, pp. 439 – 452, 2009.

[3] "Intel editorial: For self-driving cars, there's big meaning behind one big number: 4 terabytes." https://www.intc.com/investor-relations/investor-education-and-news/investor-news/press-release-details/2017/Intel-Editorial-For-Self-Driving-Cars-Theres-Big-Meaning\-Behind-One-Big-Number-4-Terabytes/default.aspx, 2017.

[4] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*. Cambridge, MA, USA: MIT Press, 1st ed., 1998.

[5] R. Bellman, "A markovian decision process," *Indiana Univ. Math. J.*, vol. 6, pp. 679–684, 1957.

[6] Z. Hameed Mir and F. Filali, "Lte and ieee 802.11p for vehicular networking: a performance evaluation," *EURASIP Journal on Wireless Comm. and Netw.*, vol. 2014, no. 1, p. 89, 2014.

[7] A. D. Shoaei, M. Derakhshani, S. Parsaeifard, and T. Le-Ngoc, "Mdp-based mac design with deterministic backoffs in virtualized 802.11 wlans," *IEEE Trans. Veh. Tech.*, vol. 65, pp. 7754–7759, Sept 2016.

[8] Q. Tse, W. Si, and J. Taheri, "Estimating contention of ieee 802.11 broadcasts based on inter-frame idle slots," in *Proc. IEEE Conf. on Local Computer Networks - Workshops*, pp. 120–127, Oct 2013.

[9] G. Bianchi, "Performance analysis of the ieee 802.11 distributed coordination function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, 2000.

[10] Z. Liu and I. Elhanany, "Rl-mac: A qos-aware reinforcement learning based mac protocol for wireless sensor networks," in *Proc. IEEE Int. Conf. on Netw., Sens. and Control*, pp. 768–773, 2006.

[11] C. Wu, S. Ohzahata, Y. Ji, and T. Kato, "A mac protocol for delay-sensitive vanet applications with self-learning contention scheme," in *Proc. IEEE Consumer Comm. and Netw. Conference*, pp. 438–443, Jan 2014.

[12] Q. Yang, S. Xing, W. Xia, and L. Shen, "Modelling and performance analysis of dynamic contention window scheme for periodic broadcast in vehicular ad hoc networks," *IET Comm.*, vol. 9, no. 11, pp. 1347–1354, 2015.

[13] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3, pp. 279–292, 1992.

[14] C. J. C. H. Watkins, *Learning from Delayed Rewards*. PhD thesis, King's College, Cambridge, UK, May 1989.